

# Application of Symbolic Computations to the Constitutive Modeling of Structural Materials

Steven M. Arnold  
*Lewis Research Center*  
*Cleveland, Ohio*

H.Q. Tan and X. Dong  
*The University of Akron*  
*Akron, Ohio*

(NASA-TM-103225) APPLICATION OF SYMBOLIC  
COMPUTATIONS TO THE CONSTITUTIVE MODELING OF  
STRUCTURAL MATERIALS (NASA) 17 6 CSCL 20K

N70-25364

Unclas  
65/30 0295191

Prepared for the  
1990 Winter Annual Meeting of the  
American Society of Mechanical Engineers  
Dallas, Texas, November 25-30, 1990





# APPLICATION OF SYMBOLIC COMPUTATIONS TO THE CONSTITUTIVE MODELING OF STRUCTURAL MATERIALS

*S.M. Arnold*

*NASA Lewis Research Center, Cleveland, Ohio 44135, U.S.A.*

*H.Q. Tan<sup>1</sup> and X. Dong*

*Department of Mathematical Sciences*

*The University of Akron, Akron, Ohio 44325, U.S.A.*

## ABSTRACT

Development of new material models for describing the behavior of real materials (e.g. soils, ceramics and metals (monolithic and composite)) represents an important area of research in various engineering disciplines. This is evidenced by research activities associated, for example, with advanced high temperature metal matrix composites, reinforced concrete and geotechnical materials, to name a few. In particular, in applications involving elevated temperatures, the derivation of mathematical expressions (constitutive equations) describing the material behavior can be quite time consuming, involved and error-prone. Therefore intelligent application of symbolic systems to facilitate this tedious process can be of significant benefit.

Presented here is a problem oriented, self contained symbolic expert system, named SDICE, which is capable of efficiently deriving potential based constitutive models in analytical form. This package, running under DOE MACSYMA, has the following features: i) partial differentiation (chain rule), ii) tensor computations (utilizing index notation) including both algebraic and calculus, iii) efficient solution of sparse systems of equations, iv) automatic expression substitution and simplification, v) back substitution of invariant and tensorial relations, vi) the ability to form the Jacobian and Hessian matrix and vii) a relational data base. Limited aspects of invariant theory have also been incorporated into SDICE due to the utilization of potentials as a starting point and the desire for these potentials to be frame invariant (objective).

The uniqueness of SDICE resides in its ability to manipulate expressions in a general yet pre-defined order and simplify expressions so as to limit expression growth. Results are displayed, when applicable, utilizing index notation. SDICE has been designed to aid and complement the human constitutive model developer. A number of examples will be utilized to illustrate the various features contained within SDICE. It is expected that this symbolic package can and will provide a significant incentive to the development of new constitutive theories.

## 1. INTRODUCTION

Efforts in constitutive research involve for example, the development of mathematical relationships for predicting reversible and irreversible material response, derivation of material stiffness matrices appropriate for finite element calculations, characterization of model parameters, formulation of a Jacobian matrix for implicit numerical integration schemes, issues of convexity, and computer implementation. The

---

<sup>1</sup>Work funded by NASA Grant NAG3-872.

entire process requires significant manual algebraic manipulations and computer programming. Hence, the response time for the related efforts is quite long. As a result, it can be rather difficult to introduce significant changes or modifications into a constitutive theory. Moreover, the outcome of the research effort may be affected by human errors which are often difficult to detect. In this regard, symbolic computation can play a major role. Furthermore, application of symbolic manipulation can provide a significant incentive to the development of new constitutive theories and their applications, e.g. finite element. Presented here is a description of a software package named SDICE (Symbolic Derivation of Constitutive Equations) which was designed to aid and complement the human constitutive model developer in constructing, analyzing and implementing potential based constitutive equations.

Results obtained from direct application of a general purpose symbolic system, such as MACSYMA (see Ref. 7), have been shown to be not useful in most cases due to the number of steps involved in the derivation process and the problems associated with expression growth (Refs. 4,5 and 17-22). For this reason, resourceful derivation procedures (which are continuously in communication with a relational data base) must be developed so that optimal results can be obtained. This is the intent behind the development of SDICE, whose uniqueness resides in its ability to manipulate expressions in a pre-defined order and simplify expressions so as to limit expression growth. The essential features of the approach taken in SDICE to address the above problems consists of:

- 1) a structured derivation procedure to avoid redundant steps and to minimize expression growth,
- 2) implementation of special procedures (e.g. procedures for simplification and pattern match) to facilitate the derivation process,
- 3) expression substitution and simplification during the entire derivation process by incorporating several levels of processing,
- 4) automatic grouping and labeling of common factors (intermediate variable substitution),
- 5) taking advantage of permutation and symmetry relationships of terms involved in each derivation step, and
- 6) as a rule-based system, intended for constitutive equation research, SDICE will record user defined rules and store them in a relational data base, whereby the information may be retrieved, redefined and restored as required.

These six features are interconnected in such a way as to allow them to work in concert with one another, thereby providing the user with simplified final expressions. In the next section a brief description of potential based constitutive equations is given. This section is followed by an overview of the special problem oriented functions presently available in SDICE as well as a section illustrating their application to a transversely isotropic formulation.

## 2. APPLICATION TO POTENTIAL BASED CONSTITUTIVE EQUATIONS

Constitutive laws provide the link between stress components  $\sigma_{ij}$  and strain components  $e_{ij}$  at any point in a body. These laws may be simple or extremely complex, depending upon the material of the body and the conditions to which it has been subjected. Consider the well known case of a hyperelastic material. Here, the stress and strain components are related through a normality structure utilizing either a strain energy or complementary energy function, i.e.

$$\sigma_{ij} = \frac{\partial W}{\partial e_{ij}} \quad (1)$$

or

$$e_{ij} = \frac{\partial \Omega}{\partial \sigma_{ij}} \quad (2)$$

For inelastic material behavior the internal state variable potential viewpoint is adopted, i.e.,

$$\Omega = \Omega(\sigma_{ij}, \alpha_\beta, T) \quad (3)$$

with the generalized normality structure

$$\dot{\epsilon}_{ij} = \frac{\partial \Omega}{\partial \sigma_{ij}} \quad i, j = 1, 2, 3 \quad (4)$$

and

$$\dot{\alpha}_\beta = -h(\alpha_\gamma) \frac{\partial \Omega}{\partial \alpha_\beta} \quad \beta, \gamma = 1, 2, \dots, n \quad (5)$$

as discussed in Refs. 10 through 12. Where  $\Omega$  is the complementary dissipation potential function,  $\epsilon_{ij}$  the inelastic strain,  $\alpha_\beta$  the internal state variables,  $n$  the number of internal

state variables and  $(\dot{\phantom{x}})$  denotes differentiation with respect to time. Equations (4) and (5) are known as the flow and evolutionary equations, respectively.

Frame invariance (objectivity) of the resulting constitutive relations is insured by requiring the potential,  $\Omega$  or  $W$ , to depend only on certain invariants and invariant products of its respective tensorial arguments, i.e., an integrity basis, see Ref. 16. Both isotropic and transversely isotropic material symmetries have been considered thus far. Transversely isotropic material symmetry is included in the potentials of Eqs. (1)–(3) by introducing a directional tensor  $u_i u_j$ , e.g.  $\Omega = \Omega(\sigma_{ij}, \alpha_\beta, u_i u_j, T)$  or  $W = W(e_{ij}, u_i u_j)$ . The symmetric tensor  $u_i u_j$  is formed by a self product of the unit vector  $u_i$  which denotes the local preferred direction.

Two viscoplastic theories have been proposed for isotropic materials, see Refs. 13 and 14, and employed to verify implementation of various special functions within SDICE as described in Refs. 1, 2 and 23. In both theories the existence of a dissipation potential is assumed, and the form is taken to be,

$$\Omega = \kappa^2 \left[ \frac{1}{2\mu} \int f(F) dF + \frac{R}{H} \int g(G) dG \right] \quad (6)$$

where the dependence of the applied stress and internal stress (cf. Eq (3)) enters through the scalar functions  $F(\Sigma_{ij})$  and  $G(a_{ij})$ , respectively.

For a material with transversely isotropic symmetry the dissipation potential is assumed to take the form of Eq. (6) where the dependence of the applied stress, internal stress and preferred direction enters through the scalar functions  $F(\Sigma_{ij}, u_i u_j)$  and  $G(a_{ij}, u_i u_j)$  respectively. The stress dependence is given by

$$F = [AJ_2 + BJ_5 + CJ_4^2] - 1 \quad (7)$$

$$G = [A\hat{J}_2 + B\hat{J}_5 + C\hat{J}_4^2] \quad (8)$$

where

$$J_2 = \frac{1}{2} \Sigma_{ij} \Sigma_{ij}$$

$$J_4 = u_i u_j \Sigma_{ji}$$

$$\begin{aligned}
J_5 &= u_i u_j \Sigma_{jk} \Sigma_{ki} \\
\hat{J}_2 &= \frac{1}{2} a_{ij} a_{ij} \\
\hat{J}_4 &= u_i u_j a_{ji} \\
\hat{J}_5 &= u_i u_j a_{jk} a_{ki} \\
\Sigma_{ij} &= S_{ij} - a_{ij} \\
S_{ij} &= \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij} \\
a_{ij} &= \alpha_{ij} - \frac{1}{3} \alpha_{kk} \delta_{ij}
\end{aligned} \tag{9}$$

Upon application of Eqs. (4) and (5) the resulting flow and evolutionary laws are:

$$\dot{\epsilon}_{ij} = \frac{f(F)}{2\mu} \Gamma_{ij} \tag{10}$$

$$\dot{a}_{ij} = h(G) \dot{\epsilon}_{ij} - \gamma(G) \pi_{ij} \tag{11}$$

respectively, where

$$\Gamma_{ij} = A \Sigma_{ij} + B[u_k u_i \Sigma_{jk} + u_j u_k \Sigma_{ki} - \frac{2}{3} J_4 u_i u_j] + \frac{2}{3} C J_4 (3u_i u_j - \delta_{ij})$$

$$\pi_{ij} = A a_{ij} + B[u_k u_i a_{jk} + u_j u_k a_{ki} - \frac{2}{3} \hat{J}_4 u_i u_j] + \frac{2}{3} C \hat{J}_4 (3u_i u_j - \delta_{ij})$$

The above transversely isotropic equations will be employed in subsequent examples to illustrate specific functions incorporated in SDICE.

### 3. SPECIAL PROBLEM ORIENTED FUNCTIONS

In deriving material constitutive equations and matrices, a number of general mathematical capabilities are desirable, including:

- 1) partial differentiation (of both scalar and tensorial functions),
- 2) tensor computations (utilizing index notation),
- 3) matrix solution of systems,
- 4) factorization of common terms,
- 5) expression substitution and simplification (i.e. intermediate scalar and tensorial variables),
- 6) back substitution of invariant and tensorial relations,

as well as some specialize capabilities such as

- 7) formation of the Jacobian and Hessian matrix
- 8) solution of eigenvalues,
- 9) the ability to input and apply simplifying conditions,
- 10) the ability to convert from index notation to matrix notation,
- 11) the automatic determination of an integrity basis (invariant theory),

- 12) the automatic fortran generation of the resulting expressions,

and last, but not least, the desire to perform these various tasks within a user friendly environment.

Symbolic packages, such as MACSYMA, MAPLE, REDUCE, MATHEMATICA, etc., have varying degrees of capability to address certain aspects of the above requirements, however, direct use of a symbolic system in the derivation process can be complicated and lead to rapid expression growth. Therefore, special purpose procedures have been and are being designed and implemented into SDICE in order to simplify and expedite the derivation process. These special procedures are implemented as separate LISP (Ref. 24) functions, thus enabling manipulations of the internal data structures of the MACSYMA expressions as well as compensation for the inefficiency and lack of simplification mechanisms within the existing MACSYMA level built-in functions.

For example our work thus far in dealing with the differentiation and the minimization of expression growth of implicit tensorial functions has resulted in the development of several strategies, namely:

- 1) Store the invariant and tensorial relations in a relational data base.
- 2) Implement procedures to compute tensor expressions according to the rules defined in tensor calculus.
- 3) Map a tensor into the domain of a scalar by utilizing a property list to store the variable and its subscript; thereby, allowing all differentiation to be treated in the same way.
- 4) Generate subscripts for intermediate tensor variables and store them in the same property list in a pre-defined order.
- 5) Represent differentiation results by a search tree, starting with the potential function as the root of the tree and its descendants without dependent relations among themselves as leaves. A separate procedure decides whether the function and its variables are tensorial or scalar.
- 6) Finally, check the property list and if the function involves tensors, subscripts are added back for the final result according to the predetermined order.
- 7) Simplification of the result is accomplished by i) checking the data base so as to replace any known invariant or intermediate relation with its corresponding name in the final expressions, ii) grouping common terms by factorization and iii) identifying and naming terms that can be written as a tensorial variable.

These strategies as well as others, are contained within a new set of problem oriented functions residing in SDICE which supplement those normally available under MACSYMA. It is important to note that these functions utilize some relevant MACSYMA functions, such as EV, RATSIMP, FACTOR, etc. However they are not merely driver routines, but functions with problem oriented computational algorithms embedded in them. A list of the name, argument list and short description of these new problem oriented functions is given in the Appendix.

#### 4. APPLICATION OF SDICE FUNCTIONS

Here a number of the available functions within SDICE will be exercised. Input required by SDICE is entered interactively through a user friendly interface and stored within a relational data base. SDICE provides the user with the ability to create, delete, modify or project a given data base as well as add, check or delete common rules, through a data base manager. Due to the fact that MACSYMA typically returns lower case letters and the desire to utilize the greek alphabet, the following notation has been adopted within SDICE:

a - z	lower case English alphabet
aa - zz	upper case English alphabet
ga - gz	lower case Greek alphabet
gaa- gzz	upper case Greek alphabet

Also note that commas between indices are merely separators and are not to be associated with spatial derivatives as standard index notation suggests.

#### 4.1 Invariant Theory

Utilization of potential functions as a starting point and the desire for these potentials to be frame invariant (objective) prompted the introduction into SDICE of limited aspects of algebraic invariant theory. As a result, given the number of symmetric second rank tensors contained within a potential and any user defined simplifying conditions, SDICE will determine automatically the corresponding integrity basis. For example, consider a function which contains two second rank symmetric tensors, defined as

$$\Sigma_{ij} = S_{ij} - a_{ij}$$

and

$$uu = u_i u_j$$

and one simplifying condition  $u_i u_i = 1$ . If we invoke SDICE by employing the function called INV,

```
in20 ==> inv([gss,uu],[uu=u[i]*u[j],u[i]*u[i]=1]);
```

the following result is obtained.

```
out20 ==> [[{gss}, gss2i, i], [{gss2}, gss2i, j gss2j, i],
           [{gss3}, gss3i, j gss3j, k gss3k, i], [{gss uu, gss uu2}, u2 gss2i, i gss2j, j],
           [{gss2 uu, gss2 uu2}, u2 gss2i, i gss2j, j k. k]]
```

Theoretically (see Ref. 16) the integrity basis for a function containing two second rank symmetric tensors should be comprised of ten invariants. However due to the special nature of uu, three invariants are found to be equal to one and are therefore discarded, while two other pairs of invariants are found to be equivalent to one another. Thus reducing the remaining seven by two. Therefore due to the imposed condition,  $u_i u_i = 1$ , the above five invariants are seen to comprise an integrity basis for this potential with these two symmetric second rank tensors. Note that brackets enclose a specific invariant denoted using index notation while sub-brackets define the invariant using matrix notation; for example the first invariant is interpreted as

$$[gss] \equiv \text{tr} \Sigma$$

in matrix notation or

$$gss_{i,j} \equiv \Sigma_{ji}$$

in index notation, while the fourth is

$$\text{tr}\Sigma u \equiv \text{tr}\Sigma u^2 \text{ or } u_i \Sigma_{ij} u_j$$

#### 4.2 Partial Differentiation

Determination of flow and evolutionary laws are directly linked to the ability to perform partial differentiation as discussed above and illustrated in Eqs. (4) and (5). This capability is accessed within SDICE through the function named DIF (see the appendix for a description of the argument list). For example considering Eqs. (4) and (6) through (9), it is apparent that in deriving the flow law for the above transversely isotropic model, the inelastic strain rate is given by:

$$\begin{aligned} \dot{\epsilon}_{ij} &= \frac{\partial \Omega}{\partial \sigma_{ij}} \\ &= \frac{\partial \Omega}{\partial F} \left( \left[ \frac{\partial F}{\partial J_2} \frac{\partial J_2}{\partial \Sigma_{kl}} + \frac{\partial F}{\partial J_5} \frac{\partial J_5}{\partial \Sigma_{kl}} + \frac{\partial F}{\partial J_4} \frac{\partial J_4}{\partial \Sigma_{kl}} \right] \frac{\partial \Sigma_{kl}}{\partial S_{rs}} \frac{\partial S_{rs}}{\partial \sigma_{ij}} \right) \end{aligned}$$

Notice that three types of partial derivatives must be taken, i) one in which both the function (numerator) and variable (denominator) are scalars, ii) one in which both the function and variable are tensors and iii) one in which the function is a scalar and the variable is a tensor. All three types are included in the DIF function. Upon employing the data base manager to store Eqs. (6) thru (9) in a data base named anij2, the above flow law may be obtained by issuing the following command.

```
in1 ==> dif(goo,gs[i,j],anij2);
```

The result returned is then:

```
out1 ==> gk2 (f(ff) (6 cc ui uj + (- 2 cc - 2 bb) gdi,j) jj4
+ f(ff) (uil (3 bb ui gssj,il + 3 bb gssi,il uj) + 3 aa gssi,j)) / (6 gm)
```

Intermediate tensorial variables can then be established by employing the function, GROUPT, as shown below;

```
in2 ==> ed[i,j]:groupt(%, [i,j]);
```

```
out2 ==> gk2 (f(ff) (6 cc ui uj + (- 2 cc - 2 bb) gdi,j) jj4
+ f(ff) (3 bb ggt1i,j + 3 aa gssi,j)) / (6 gm)
```

```
in3 ==> ggt1[i,j];
```

```
out3 ==> ui uil gssj,il + gssi,il uil uj
```

Here only one level of tensor substitution (intermediate variable substitution) is invoked. Comparing the expression denoted by out2 above with Eq. (10) we see, after minor simplification, that they are equivalent. This final factoring out of common terms such as  $f(ff)$  and finding the minimum common denominator is still a topic of ongoing research. Additional details regarding the differentiation algorithm can be found in Refs. 2 and 23.

#### 4.3 Eigenvalue Problem

A question of importance to constitutive developers is whether or not the flow (or yield) surface one is constructing or employing is convex. One way to ascertain this is to calculate the corresponding eigenvalues and examine their sign. Here we will determine the eigenvalues corresponding to the function  $F$ , see Eq. (7). This example was selected based on the fact that numerous SDICE functions are utilized, e.g. INPCON, GETF, APPCON, RATSMP\_A, FORMM and CHAR.

Let us begin by imposing specific conditions on the function  $F$ , i.e. assume a principle stress state, take the internal stress state to be zero, and examine a specific direction orientation,  $u_i$ . This is easily accomplished by invoking the function INPCON.

```
in5 ==> inpcon(1);

TEST 1

PLEASE INPUT THE CONDITIONS (type d or done when finished) ...

ENTER THE CONDITION : gs[i,j]=[(p,0,0],[0,p,0],[0,0,p]];
ENTER THE CONDITION : ga[i,j]=[[0,0,0],[0,0,0],[0,0,0]];
ENTER THE CONDITION : u[i]=[1,0,0];
ENTER THE CONDITION : d;

out5 ==> done
```

Equation (7) is retrieved from the data base,  $anj2$ , by invoking the GETF function.

```
in6 ==> getf(ff,anj2);

out6 ==> bb jj5 + cc jj42 + aa jj2 - 1
```

The set of conditions previously entered with the INPCON command are then applied to the above expression using the APPCON function.

```
in7 ==> appcon([1],%);

eqn1 : ((gs23,3 + (2 gs22,2 - 4 gs21,1) gs23,3 + gs22,2 - 4 gs21,1 gs22,2
+ 4 gs21,1) cc + (gs23,3 + (2 gs22,2 - 4 gs21,1) gs23,3 + gs22,2
- 4 gs21,1 gs22,2 + 4 gs21,1) bb + (3 gs23,3 + (- 3 gs22,2 - 3 gs21,1) gs23,3
+ 3 gs22,2 - 3 gs21,1 gs22,2 + 3 gs21,1) aa - 9)/9
```

```
out7 ==> [eqn1]
```

Due to the scalar nature of F, see result out6, only a single equation is returned. If the expression of interest were tensorial then multiple equations could be returned depending upon the imposed conditions. The diagonal components  $\sigma_{11}, \sigma_{22}$  and  $\sigma_{33}$  are then replaced by their principal stress representations  $\sigma_1, \sigma_2$  and  $\sigma_3$  respectively, by employing the alternative form of the APPCON function.

```
in8 ==> appcon(eqn1, [gs[1,1]=gs1,gs[2,2]=gs2,gs[3,3]=gs3]);
```

```
out8 ==> ((cc + bb + 3 aa) gs32 + ((2 cc + 2 bb - 3 aa) gs22
+ (- 4 cc - 4 bb - 3 aa) gs1) gs3 + (cc + bb + 3 aa) gs22
+ (- 4 cc - 4 bb - 3 aa) gs1 gs2 + (4 cc + 4 bb + 3 aa) gs12 - 9)/9
```

The nonphysical coefficients A,B, and C are now replaced by their physically meaningful counterparts,  $\kappa_T, \eta$  and  $\omega$  (denoted by gk, ge, go, respectively) determined elsewhere, by again utilizing the function APPCON.

```
in9 ==> appcon(%,{aa=1/gk^2,bb=-(ge^2-1)/(ge^2*gk^2),
cc=1/gk^2*((2+go^2)/(4*ge^2-1)-1/ge^2)});
out9 ==> (- 4 gk go2 + gs3 (gs2 (1 - 2 go2) - gs1) + go2 gs32 + go2 gs22
- gs1 gs2 + gs12 + gk )/(4 gk go2 - gk2)
```

The resulting expression is then simplified utilizing the RATSMP\_A procedure. This function differs from MACSYMA's simplification function RATSIMP in two primary ways. First, it allows a priority by variable or expression to be specified during the simplification process, thereby giving the user control over the appearance of the final expression. Secondly it performs intermediate variable substitution operations throughout the simplification process, thus resulting in more compact and tractable final expressions.

```
in10 ==> ratsmp_a(%,[gs1,gs2,gs3],1);
```

$$qq1 = \frac{go^2}{gk (2 go - 1) (2 go + 1)}$$

$$qq2 = \frac{2 go^2 - 1}{gk (2 go - 1) (2 go + 1)}$$

$$qq3 = \frac{1}{gk^2 (2 go - 1) (2 go + 1)}$$

$$\begin{aligned} out10 ==> & qq1^2 gs3^2 + qq1^2 gs2^2 + qq3^2 gs1^2 - gs1 gs3 qq3 - gs1 gs2 qq3 \\ & - gs2 gs3 qq2 - 1 \end{aligned}$$

At this stage one can either employ the function EIGEN, described in the appendix, or make use of the various procedures utilized by EIGEN to perform the task. In order to give the reader a clearer understanding of the various features of SDICE, the latter option has been selected.

The following quadratic polynomial can be written in matrix form by utilizing the FORMM function, that is

```
in11 ==> formm(%, {gs1,gs2,gs3});
```

```
out11 ==> x aa x - 1
           i i, j j
```

where

```
in12 ==> x[i];
```

```
out12 ==> [ gs1 gs2 gs3 ]
```

```
in13 ==> aa[i,j];
```

```
out13 ==> [ [ qq3 - qq3 qq3 ]
            [ qq3 - 2 - 2 ]
            [ [ qq3 qq2 ]
              [ - 2 qq1 - 2 ]
              [ [ qq3 qq2 ]
                [ - 2 - 2 qq1 ]
                [ [ 2 2 ]
```

```
in14 ==> x[j];
```

```
out14 ==> [ gs1 ]
           [ [ gs2 ]
           [ [  ]
           [ [ gs3 ]
```

The characteristic equation is then determined using the CHAR procedure.

```
in15 ==> char(aa[i,j]);
```

$$\begin{aligned} out15 ==> & - (-2 ggz + qq2 + 2 qq1) (-2 ggz^2 + qq3^2 + 2 ggz qq3 + qq2 qq3 \\ & - 2 qq1 qq3 - ggz qq2 + 2 ggz qq1)/4 \end{aligned}$$

where  $ggz$  represents the typical eigenvalue  $\lambda$ . The characteristic polynomial is then simplified using intermediate expressions as follows:

```
in16 ==> ratsmp_a(%,ggz,1);

qq4 = (qq2 + 2 qq1) qq3 (qq3 + qq2 - 2 qq1)

qq5 = 2 qq32 - 8 qq1 qq3 + qq22 - 4 qq12

qq6 = qq3 + 2 qq1
```

```
out16 ==> - ggz3 + ggz2 qq6 +  $\frac{ggz qq5}{4}$  -  $\frac{qq4}{4}$ 
```

```
in17 ==> appcon(qq4);
```

```
out17 ==> 0
```

```
in18 ==> out16+'qq4/4;
```

```
out18 ==> - ggz3 + ggz2 qq6 +  $\frac{ggz qq5}{4}$ 
```

The eigenvalues are finally determined by using the MACSYMA SOLVE function, i.e.

```
in19 ==> solve(%,ggz);
```

```
out19 ==> [ggz =  $\frac{qq6 - \sqrt{qq6^2 + qq5}}{2}$ , ggz =  $\frac{\sqrt{qq6^2 + qq5} + qq6}{2}$ , ggz = 0]
```

Inequalities can then be generated, if necessary, to restrict the material parameters so that the surface is convex. Development of procedures to assist in the establishment of these inequalities is an extremely challenging task and is an area of current research.

## 5. CONCLUSION

A problem oriented, self contained symbolic expert system, named SDICE, which runs under DOE MACSYMA and is capable of efficiently deriving a special class of constitutive equations in analytical form, namely those derivable from a potential function, has been presented. A new set of special purpose procedures have been developed which provide capabilities in areas such as; partial differentiation (chain rule) of both scalars and tensorial functions, efficient solution of sparse systems of equations, automatic expression substitution and simplification, back substitution of invariant and tensorial relations, formation of Jacobian and Hessian matrices, and algebraic invariant theory. A number of examples have been included to illustrate various features contained within SDICE.

The uniqueness of SDICE resides in its ability to simulate the human intelligence required to manipulate expressions in a general yet pre-defined order and simplify

expressions so as to limit expression growth. This simulation is accomplished through the utilization of a relational data base, in that all problem-oriented algorithms are continuously in communication with this relational data base thus allowing automatic expression and variable substitution throughout the solution process.

Work is continuing, with emphasis upon 1) further improvement in the simplification capabilities available, 2) in a user friendly interactive interface, 3) automatic Fortran code generation of the resulting expressions and 4) automation of the identification of inequalities. In summary the application of intelligent symbolic methods in the development, analysis and implementation of constitutive theories has thus far shown widespread benefits in expediting and reducing the error-prone nature of this tedious process. Public release of SDICE is anticipated in the near future.

## REFERENCES

1. Arnold S.M., H.Q. Tan, and X. Dong, "Computer Simulation of the Mathematical Modeling Involved in Constitutive Equation Development: Via Symbolic Computations", NASA TM 102532, Aug 1989.
2. Arnold, S.M. and Tan, H. Q., "Symbolic Derivation of Potential Based Constitutive Equations", Computational Mechanics, Vol 6, 1990.
3. Cecchi, M. M. and Lami, C., "Automatic generation of stiffness matrices for finite element analysis", Int. J. Num. Meth. Eng., 11, 1977, pp.396-400.
4. Chang, T.Y., Saleeb, A.F., Wang, P.S. and Tan, H.Q., "On the Symbolic Manipulation and Code Generation for Elasto-Plastic Material Matrices", Engineering with Computers, Vol 1, 1986, pp 205-215.
5. Chang, T.Y., Tan, H.Q., and Zheng, D., "Application of Symbolic Methods to Hybrid/Mixed Finite Elements", Computers and Structures, Vol 10., No. 1, 1990.
6. Golden, J.P., "Differentiation of Unknown Functions in MACSYMA", ACM Bulletin, Vol 19, No. 2, May 1985.
7. MACSYMA, "MACSYMA Reference Manual", Version 10, The MATHLAB Group, Laboratory for Computer Science, MIT, 1984.
8. Noor, A.K. and Andersen, C.M., "Computerized Symbolic Manipulation in Nonlinear Finite Element Analysis", Computer and Structures, Vol 1, 1981, pp. 9-40.
9. Noor, A.K., Andersen, C.M., "Computerized Symbolic Manipulation in Structural Mechanics - Progress and Potential", Computer Structures, Vol. 10, 1977, pp. 95-118.
10. Ponter, A. R. S., "Convexity and Associated Continuum Properties of a Class of Constitutive Relationships", Journal de Mecanique, Vol 15, No. 4, 1976, pp. 527-542.
11. Ponter, A.R.S., Leckie, F.A., "Constitutive Relationships for Time-Dependent Deformation of Metals", J. Eng. Mech. Technol., Vol 98, No. 1, 1976, pp 47-57.
12. Rice, J.R., "On the Structure of Stress-Strain Relations for Time Dependent Plastic Deformations in Metals", J. Appl. Mech., Vol 37, No. 3, 1970, pp 728-737.
13. Robinson, D.N., "A Unified Creep-Plasticity Model for Structural Metals at High Temperature", ORNL/TM5969, Oak Ridge National Laboratory, 1978.
14. Robinson, D.N., "A Multiaxial Theory of Viscoplasticity for Isotropic Materials", Turbine Engine Hot Section Technology 1986, NASA CP2444, 1986, pp. 283-292.

15. Robinson, D.N., Duffy, S.F., and Ellis, J.R., "A ViscoPlastic Constitutive Theory for Metal Matrix Composites at High Temperature", The 1987 Pressure Vessels and Piping Conference, PVP-Vol 123, June 28 – July 2, 1987, pp. 49–56.
16. Spencer, A.J.M., "Theory of Invariants", In Continuum Physics, Ed. Eringen A.C., Academic Press, Vol I, 1971, pp. 240–355.
17. Tan, H.Q., "Integration of Symbolic and Numerical Computations in Finite Element Analysis", IMACS Transactions on Scientific Computing, Vol. 1: Numerical and Applied Mathematics, J.C. Baltzer AG, Scientific Publishing Company, Basel, Switzerland, pp. 641–645.
18. Tan, H.Q., Chang T.Y. and Zheng D., "On Symbolic Manipulation and Code Generation of a Hybrid 3-D Solid Element", to appear, Journal of Engineering with Computers, Springer-Verlag, New York.
19. Tan, H.Q., "Automatic Equation Derivation and Code Generation in Engineering Analysis", Proceedings of IEEE International Computer Science Conf. '88 (Artificial Intelligence: Theory and Practice., Hong Kong, 1988.
20. Tan, H. Q., "Symbolic Derivation of Equations for Mixed Formulation in Finite Element Analysis", Computers and Mathematics, Springer-Verlag, 1989, pp. 289–298.
21. Tan, H.Q., "Integration of Symbolic and Numerical Computations in Finite Element Analysis", Proceedings of 12th IMACS World Congress on Scientific Computation, Paris, France, 1988.
22. Tan, H.Q., "Symbolic Derivation of Material Property Matrices in Finite Element Analysis", ASME, HTD-Vol. 105, 1988, pp. 111–116.
23. Tan, H.Q., Arnold, S.M. and Dong, X., "Symbolic Derivation of Constitutive Equations ", ASME, HTD-Vol. 105, 1988, pp. 103–110.
24. Wilensky, R., "LISPcraft", W.W. Norton Co, 1984.

## APPENDIX – LIST OF AVAILABLE SDICE FUNCTIONS

APPCON([n1,n2,...],obj)

Applies [n1,n2,...] test conditions to the object obj, where obj can be either an atom or a tensor with appropriate indicies.

Alternate form

APPCON([eqn1,eqn2,...],[conditions])

Simplifies eqn1,eqn2,etc. corresponding to the specified conditions given in the second argument list.

CHAR(m)

Calculates the characteristic polynomial of the matrix m. Intermediate variables are used to simplify the results.

DIF(fn1,fn2,dbn)

This function differentiates fn1 with respect to fn2, where fn1 or fn2 may be either a scalar or tensor expression. If fn1 is an implicit function, DIF will automatically apply the chain rule to the extent required. Similarly, the form of the result is consistent with the initial form of fn1 and fn2 defined in the data base. Both fn1 and fn2 are defined in the data base dbn.

EIGEN(exp,var1,var2,...)

Calculates the eigenvalues of the matrix formulated from the given quadratic expression, exp, with dependencies var1,var2,etc. Combines functions FORMM and CHAR into one command.

FORMM(exp,[var1,var2,...])

Formulates the matrix form of the expression, exp. Intermediate variables are used to simplify the results.

GET\_LEAD\_M(m,s)

Obtains the leading principle matrix m with dimensions s by s.

GETF(fn,dbn)

Obtains the definition of the function fn defined in the data base dbn.

GROUPT(exp,[k,l])

Groups tensor product terms, possessing common subscripts [k,l] and automatically generates a tensor intermediate variable  $ggti_{k,l}$  which represents these terms.

HESS([fn1,fn2,...],[var1,var2,...],dbn)

Calculates the Hessian matrix. Note fn1,fn2,etc. are defined in the data base dbn with var1,var2,etc. as their dependents.

INPCON(n)

Allows user to enter n sets of test conditions, which will be used by the APPCON function. These conditions remain available until one exits from the execution sub-level of SDICE.

INV([ten1,ten2,...],[conditions...])

Automatically derives the integrity basis for a given function, provided the tensors (ten1,etc.) are symmetric and of rank two. The second argument can contain the definition of the tensors as well as any other condition or simplification rules one wants applied.

JAC([fn1,fn2,...],[var1,var2,...],dbn)

Calculates the Jacobian matrix. Note fn1,fn2,etc. are defined in the data base dbn with var1,var2,etc. as their dependents.

**LINSOLVE2([eqn1,eqn2,...],[var1,var2,...])**  
 Solves the system of linear equations simultaneously for the list of variables. This function is significantly more efficient than the existing package under MACSYMA in handling sparse systems.

**PUTF(exp,dbn)**  
 Stores the expression exp as the definition of a function (named by the user) in the data base dbn.

**RATSMP(exp, v1,v2,...,vn)\***  
 Enables rational simplification of the expression exp relative to the specification of the variable (function or expression) vn. The order is important since v1 has the highest priority and vn the lowest. If a variable within exp is missing then this variable is given lower priority than the rightmost vn.

**RATSMP\_A(exp,v1,v2,...,vn,flag)\***  
 Enables rational simplification of exp relative to the priority of v1,v2,...,vn, respectively. However with this function call intermediate variables are generated to represent sub expressions of variables of lower priority and are displayed if flag is 1.

\* Note in both RATSMP functions multiple variables can be given the same priority by enclosing within brackets.

**SETDBN(dbn)**  
 Identifies dbn as the working data base.

**SETPRI(var1,var2,...varn)**  
 Defines the global priority of variables, with respect to simplification routines. Var1 has the highest priority while the others follow sequentially.

**SHOWPRI**  
 Shows the current priority list.

**SOLVE2([eqn1,eqn2,...],[var1,var2,...])**  
 Solves the list of simultaneous polynomial equations (linear or nonlinear) for the list of variables. It utilizes the SOLVE function provided by MACSYMA as well as the simplification functions provided in SDICE.

**VWCON(n)**  
 Displays the nth set of test conditions.

# Report Documentation Page

1. Report No. NASA TM-103225		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Application of Symbolic Computations to the Constitutive Modeling of Structural Materials				5. Report Date	
				6. Performing Organization Code	
7. Author(s) Steven M. Arnold, H.Q. Tan, and X. Dong				8. Performing Organization Report No. E-5641	
				10. Work Unit No. 510-01-1A	
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191				11. Contract or Grant No.	
				13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001				14. Sponsoring Agency Code	
15. Supplementary Notes Prepared for the 1990 Winter Annual Meeting of the American Society of Mechanical Engineers, Dallas, Texas, November 25-30, 1990. Steven M. Arnold, NASA Lewis Research Center. H.Q. Tan and X. Dong, Department of Mathematical Sciences, The University of Akron, Akron, Ohio 44325. H.Q. Tan's work was funded by NASA Grant NAG3-872.					
16. Abstract Development of new material models for describing the behavior of real materials (e.g. soils, ceramics and metals (monolithic and composite)) represents an important area of research in various engineering disciplines. This is evidenced by research activities associated, for example, with advanced high temperature metal matrix composites, reinforced concrete and geotechnical materials, to name a few. In particular, in applications involving elevated temperatures, the derivation of mathematical expressions (constitutive equations) describing the material behavior can be quite time consuming, involved and error-prone. Therefore intelligent application of symbolic systems to facilitate this tedious process can be of significant benefit. Presented here is a problem oriented, self contained symbolic expert system, named SDICE, which is capable of efficiently deriving potential based constitutive models in analytical form. This package, running under DOE MACSYMA, has the following features: i) partial differentiation (chain rule), ii) tensor computations (utilizing index notation) including both algebraic and calculus, iii) efficient solution of sparse systems of equations, iv) automatic expression substitution and simplification, v) back substitution of invariant and tensorial relations, vi) the ability to form the Jacobian and Hessian matrix and vii) a relational data base. Limited aspects of invariant theory have also been incorporated into SDICE due to the utilization of potentials as a starting point and the desire for these potentials to be frame invariant (objective). The uniqueness of SDICE resides in its ability to manipulate expressions in a general yet pre-defined order and simplify expressions so as to limit expression growth. Results are displayed, when applicable, utilizing index notation. SDICE has been designed to aid and complement the human constitutive model developer. A number of examples will be utilized to illustrate the various features contained within SDICE. It is expected that this symbolic package can and will provide a significant incentive to the development of new constitutive theories.					
17. Key Words (Suggested by Author(s)) Symbolic computations; Constitutive modeling; Expert system; MACSYMA; Relational data base				18. Distribution Statement Unclassified - Unlimited Subject Category 39	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 16	
				22. Price* A03	